BEYOND EXPERT TRAJECTORY MATCHING: GEOMETRY AND RISK-AWARE IMITATION LEARNING FOR SAFER UAV VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

UAV vision-language navigation (UAV-VLN) requires UAVs to follow natural-language instructions while avoiding obstacles using visual observations. A common approach to UAV-VLN is imitation learning, where navigation models are trained to mimic actions from expert demonstrations. However, vanilla imitation learning cannot distinguish multiple expert trajectories in terms of task-relevant qualities, and one particularly important quality for UAV navigation is safety. In reality, expert trajectories include ones that remain comfortably in open spaces, and ones that pass dangerously close to obstacles. Regardless of the difficulties in obstacle avoidance, vanilla imitation learning treats these trajectories as equally desirable supervision, which is not helpful for lowering collision rates. In order to address this, we adopt a simple but effective approach named GeoRisk-IL. Specifically, we introduce a collision-risk penalty on predicted actions using the distance between future UAV locations and obstacles. An essential aspect of our design is to also enhance the 3D awareness of the UAV-VLN model. This provides the spatial understanding needed for meaningful risk estimation. Interestingly, the collision-risk penalty becomes more effective when combined with geometric representation alignment, despite the latter providing little to no benefit on its own, highlighting a strong synergy between geometric awareness and risk-aware learning. On both synthetic and real-world datasets, the proposed method demonstrates significant improvements over previous methods, verifying its effectiveness. Code is available at `https://github.com/AnonymousSubmission2222/georisk`.

1 INTRODUCTION

Vision-Language Navigation (VLN) requires an embodied agent to follow natural-language instructions based on visual observations (Anderson et al., 2018b; Krantz et al., 2020; Zhang et al., 2024). Extending VLN to Unmanned Aerial Vehicles (UAVs) introduces additional challenges, as UAVs navigate in continuous three-dimensional environments and require accurate motion prediction for reliable flight (Fan et al., 2023). Most existing methods train navigation models to mimic expert trajectory demonstrations following the imitation learning paradigm (Liu et al., 2023; Wang et al., 2025b; Lee et al., 2025).

However, expert trajectories can differ substantially in their safety margins. As illustrated in Figure 1a, some trajectories stay in open spaces and leave ample distance to obstacles. In contrast, others leave a very small obstacle clearance, where even a minimal deviation from the expert path may crash the drone (Figure 1b). Standard imitation learning treats every demonstrated trajectory as an equally valid target, and does not account for their safety margins at all (Hussein et al., 2017). Thus, blindly following these expert demonstrations cannot enhance the collision avoidance ability of UAVs. In this paper, we set forth to investigate other supervision signals that explicitly account for the collision risk of predicted motions.

In this paper, we propose GeoRisk-IL (Geometry- and Risk-Aware Imitation Learning), a simple-yet-effective framework that considers both geometry representation alignment and collision-risk penalty. First, we introduce a collision-risk penalty computed from the distance between predicted

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

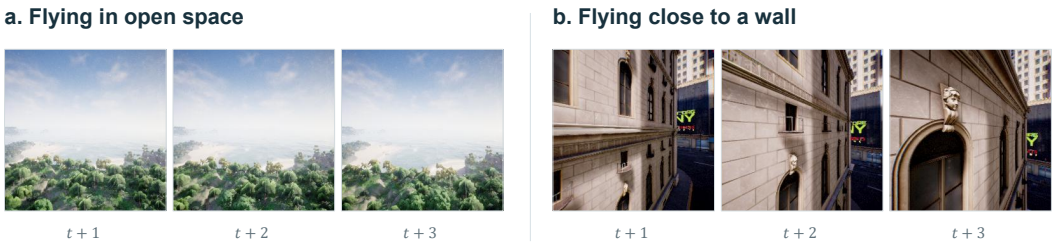

Figure 1: Expert trajectories with different obstacle-avoidance difficulties. (a) Flying through open space in TropicalIsland. (b) Flying close to a building wall in NewYorkCity.

future UAV positions and surrounding obstacles. Predictions with insufficient clearance receive larger penalties, while the penalty weakens once the clearance is sufficient. It encourages safer motions beyond simply imitating the demonstrated path, while retaining expert trajectory supervision for instruction-following behavior.

On top of this, to better enable learning from this risk-aware supervision, we also enhance the navigation model's 3D awareness using geometry representation alignment. Learning to avoid hazardous predictions requires understanding of how these motions relate to nearby obstacles in three-dimensional space. To achieve this, we align the VLM backbone outputs (Bai et al., 2025) with geometric features from a pretrained geometry foundation model (Lin et al., 2025). This representation-level supervision encourages the navigation model to capture spatial structure relevant to the visual cues and obstacle avoidance, complementing the action-level collision-risk penalty.

Our training objective includes expert trajectory supervision, geometric representation alignment, and the collision-risk penalty, jointly promoting instruction-following behavior, 3D spatial understanding, and safe motion prediction. Crucially, adding collision-risk supervision on top of geometric alignment yields larger gains than adding it to vanilla imitation learning, even though geometric alignment alone provides little to no improvement. This demonstrates the synergy between 3D spatial understanding and collision awareness, and further validates our design.

We evaluate GeoRisk-IL on the synthetic OpenUAV benchmark (Wang et al., 2025b) and the real-world collected HUGE-Bench (Guo et al., 2026). The results show substantial improvements in navigation performance and fewer collisions. The auxiliary geometric processing and collision-risk computation are used only during training, introducing no additional inference-time modules.

2 Related Works

2.1 Imitation Learning for UAV Navigation

UAV navigation has increasingly adopted imitation learning to learn continuous flight behavior from expert demonstrations (Osa et al., 2018; Blukis et al., 2018; Liu et al., 2023; Wang et al., 2025b; 2026). TravelUAV learns hierarchical trajectory prediction from human demonstrations and further augments training data through backtracking-based data aggregation (Wang et al., 2025b). UAV-Flow formulates language-conditioned fine-grained UAV control as imitation learning from expert pilot trajectories (Wang et al., 2026). SINGER instead generates collision-free expert trajectories with a planner and trains an onboard visuomotor model to imitate the resulting flight behavior (Adang et al., 2025). Recent works further improve demonstration-based learning from different aspects: LongFly incorporates historical observations and trajectory context for long-horizon waypoint prediction (Jiang et al., 2025), while DynFly learns continuous trajectory generation in B-spline space with additional supervision on UAV motion characteristics (Jiang et al., 2026b). FlightGPT also uses supervised demonstrations to initialize the navigation model before subsequent reinforcement learning (Cai et al., 2025).

These approaches improve imitation through demonstration collection and generation, temporal context, trajectory modeling, or subsequent fine-tuning with reinforcement learning. Our focus is complementary: trajectory matching alone does not explicitly measure the clearance of predicted motions from surrounding obstacles. Even when an expert demonstration is collision-free, the model's

prediction may deviate from it and lose the original safety margin. GeoRisk-IL therefore augments expert trajectory supervision with a distance-based collision-risk penalty computed from the model's own predicted trajectory.

2.2 Geometric Representation Learning for Embodied Navigation

Geometric representations have been widely used to provide embodied agents with structured spatial information (Wang et al., 2021; Chen et al., 2022). VLMaps fuses visual-language features with 3D reconstruction to build spatial maps for language-conditioned navigation (Huang et al., 2023), while ConceptFusion integrates open-set multimodal features into dense 3D maps (Jatavallabhula et al., 2023). For VLN, VER aggregates multi-view observations into a volumetric representation and learns geometric properties such as occupancy, room layout, and object structure (Liu et al., 2024). HNR-VLN uses a pretrained neural radiance representation to predict future environmental features for lookahead navigation (Wang et al., 2024). MapNav represents navigation history using an annotated top-down semantic map (Zhang et al., 2025), while MapDream learns a compact BEV representation directly from navigation objectives (Lian et al., 2026). More recently, GA-VLN combines depth-based BEV projection with features from a pretrained 3D foundation model (Yang et al., 2026), and SpatialFly injects and aligns VGGT geometric features with visual representations for UAV VLN (Jiang et al., 2026a; Wang et al., 2025a).

These methods mainly use geometry to improve scene representations, spatial reasoning, and navigation context. Geometry can also provide direct supervision for collision avoidance by measuring how closely a predicted flight path approaches surrounding obstacles. This use of geometry to supervise predicted UAV motions has received less attention. GeoRisk-IL uses a pretrained geometry model to supervise navigation representations and local obstacle geometry to penalize predicted trajectories with insufficient clearance. Many existing geometry-enhanced methods retain geometric maps or feature extractors during inference. In GeoRisk-IL, the additional geometric supervision is confined to training, preserving RGB-only navigation without additional inference-time computation.

3 GeoRisk-IL

3.1 Overall Framework

As shown in Figure 2, GeoRisk-IL builds on a Qwen3-VL-2B-Instruct (Bai et al., 2025) backbone augmented with LoRA (Hu et al., 2021) adapters and task-specific prediction heads. During training, the VLM backbone is frozen, while only the LoRA adapters and task-specific heads are optimized.

At each navigation step t, the VLM takes the RGB observations I_t, language instruction ℓ, and UAV state s_t as input. A trajectory head predicts K future UAV positions from the hidden state of the <traj> token, $\hat{W}_t = (\hat{w}_t^1, \ldots, \hat{w}_t^K)$. The predicted trajectory is supervised by the expert trajectory $W_t^* = (w_t^{1,*}, \ldots, w_t^{K,*})$ with a Smooth L1 trajectory-matching loss $\mathcal{L}_{\text{traj}}$ (Girshick, 2015). The backbone also outputs a hidden representation for a dedicated <stop> token. An MLP stop head uses this representation to predict the stopping probability, with supervision and the stopping criterion described in Appendix A.4.

Beyond expert trajectory matching, GeoRisk-IL directly supervises the collision risk of predicted motions during training. A collision-risk penalty $\mathcal{L}_{\text{risk}}$ evaluates the predicted trajectory against local obstacle geometry and penalizes predictions with insufficient clearance. Geometric representation alignment $\mathcal{L}_{\text{geo}}$ provides spatial supervision for the RGB features used in navigation, helping the model learn obstacle avoidance from the collision-risk penalty. The collision-risk penalty acts on predicted trajectories, whereas geometric alignment supervises the shared representations used for navigation.

The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \lambda_{\text{stop}}\mathcal{L}_{\text{stop}} + \lambda_{\text{risk}}\mathcal{L}_{\text{risk}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}}, \tag{1}$$

where λ_{stop}, λ_{risk}, and λ_{geo} are the weights of the stopping loss, collision-risk penalty, and geometric alignment loss, respectively. The geometry teacher, alignment head, and auxiliary obstacle-geometry and collision-risk computations are used only during training. At inference, the model predicts trajectories without these auxiliary components.

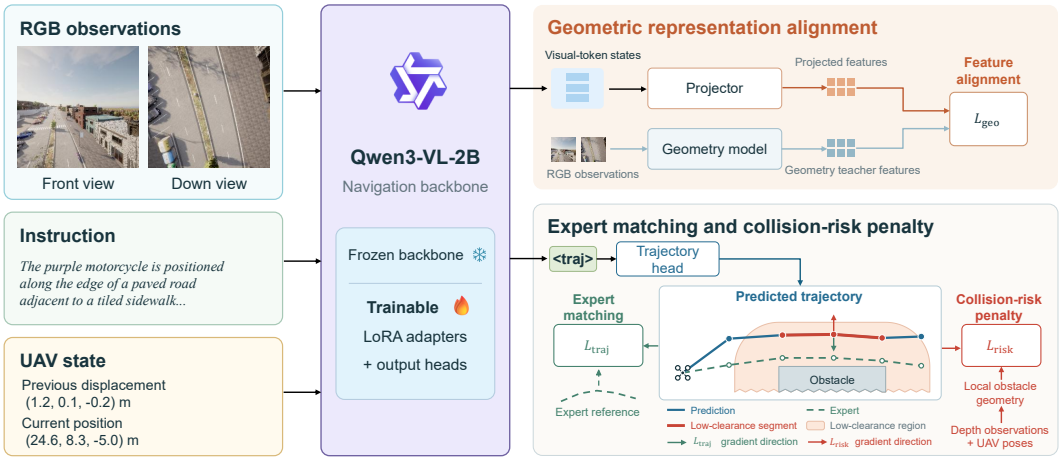

Figure 2: Overview of GeoRisk-IL. The figure shows three supervision branches: a) expert trajectory matching, b) geometric representation alignment, and c) collision-risk supervision. The geometry foundation model and collision-risk computation are used only during training, introducing no extra modules or inference-time overhead at test time.

3.2 Geometry-Aware Representation Learning

We use a frozen Depth Anything 3 (DA3) model (Lin et al., 2025) to provide geometric supervision for the navigation backbone's visual-token representations. Given the RGB observations I_t, DA3 processes the views jointly and extracts a spatial feature map for each view. The teacher and student receive the same RGB views. We use average pooling to reduce each teacher feature map to the same grid size as the student's visual tokens. Each visual-token state is then aligned with the teacher feature at the corresponding position in the same view.

We omit the time index in the feature notation for brevity. For each view v and spatial position i, let $\mathbf{g}_{v,i} \in \mathbb{R}^{d_g}$ denote the corresponding DA3 feature, and let $\mathbf{h}_{v,i}^{\text{vlm}} \in \mathbb{R}^{d_h}$ denote the final-layer VLM hidden state at the corresponding visual-token position. Here, d_g denotes the dimensionality of the DA3 spatial features, and d_h denotes the hidden-state dimensionality of the VLM backbone. This attaches geometric supervision to the VLM backbone's internal representations, alongside the trajectory prediction objective.

We use an alignment head $P_\phi : \mathbb{R}^{d_h} \to \mathbb{R}^{d_g}$ to project the VLM representation into the teacher feature space. Both student and teacher features are ℓ_2-normalized:

$$\hat{\mathbf{z}}_{v,i} = \frac{P_\phi(\mathbf{h}_{v,i}^{\text{vlm}})}{\|P_\phi(\mathbf{h}_{v,i}^{\text{vlm}})\|_2}, \qquad \hat{\mathbf{g}}_{v,i} = \frac{\mathbf{g}_{v,i}}{\|\mathbf{g}_{v,i}\|_2}. \tag{2}$$

Normalization removes differences in feature magnitude between the two networks. The squared distance between unit-length features is proportional to cosine distance, making alignment depend on feature direction.

We then align features at corresponding views and spatial positions:

$$\mathcal{L}_{\text{geo}} = \frac{\sum_{v,i} a_{v,i} \, \|\hat{\mathbf{z}}_{v,i} - \text{sg}(\hat{\mathbf{g}}_{v,i})\|_2^2}{\sum_{v,i} a_{v,i}}, \tag{3}$$

where $a_{v,i}$ indicates whether a visual-token position is valid for alignment, and $\text{sg}(\cdot)$ stops gradients through the teacher features. The loss is averaged over valid visual-token positions.

The geometry loss trains the alignment head and supervises the hidden states at visual-token positions, while the trajectory head reads the hidden state of the `<traj>` token. Both objectives update the shared backbone through its LoRA adapters. Geometric supervision can therefore influence trajectory learning through these shared parameters. At inference, the model predicts trajectories from RGB observations without running the geometry teacher or alignment head.

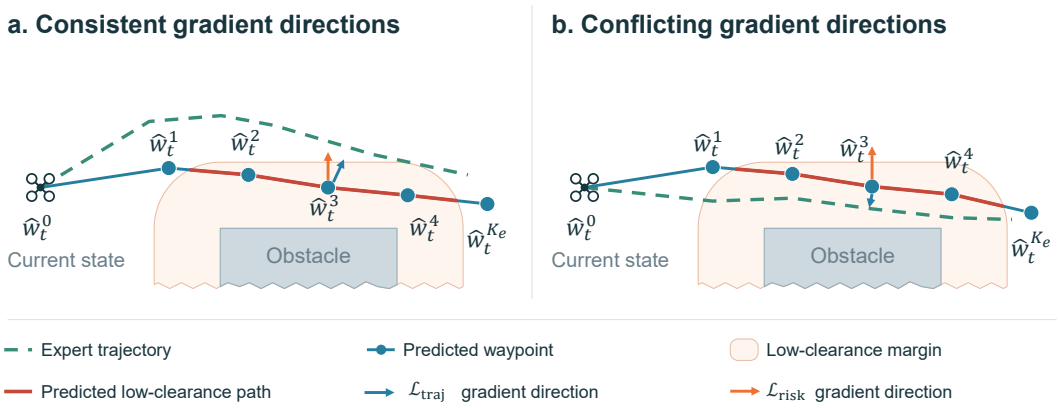

Figure 3: Effect of the imitation loss and the collision-risk penalty on predicted future waypoints. (a) In scenarios where the predicted waypoints are too close to the obstacles, while the expert path remain in safe clearance, both losses move the prediction away from the obstacle. (b) When expert demonstrations are dangerously close to obstacles, the collision-risk penalty operates against the traditional imitation loss to promote safe behavior.

3.3 Collision-Risk Supervision

We compute the collision-risk penalty from the distance between the predicted trajectory and nearby obstacles. Depth images from the current and recent steps are back-projected into 3D and transformed into the current UAV coordinate frame using the corresponding poses. After voxelization, these points form the local obstacle set $\mathcal{P}_t$.

The model predicts K trajectory points at each navigation step. The UAV follows the first $K_e \leq K$ points, takes a new observation, and predicts the next trajectory. We denote the path from its current position $\hat{w}_t^0$ through these trajectory points by $\hat{\mathcal{T}}_t^{\text{exec}}$. We apply the collision-risk penalty to this path because it is the part the UAV will actually follow before replanning. At a position $\mathbf{x}$ on this path, the clearance is

$$c_t(\mathbf{x}) = \min_{p \in \mathcal{P}_t} \|\mathbf{x} - p\|_2 - \frac{h}{2}, \tag{4}$$

where h is the voxel size. The position $\mathbf{x}$ denotes the UAV center, and p is an occupied voxel center. The subtraction of $h/2$ corresponds to an isotropic expansion around each center, yielding an approximate distance to the represented obstacle. We use a safety margin m to encourage the UAV to stay at least this far from obstacles. Predictions with clearance below m receive a stronger penalty.

The collision-risk penalty averages the following expression along the path:

$$\mathcal{L}_{\text{risk}} = \operatorname*{avg}_{\mathbf{x} \in \hat{\mathcal{T}}_t^{\text{exec}}} \left[\frac{\tau}{m} \log \left(1 + \exp \left(\frac{m - c_t(\mathbf{x})}{\tau} \right) \right) \right], \tag{5}$$

Here, avg denotes the mean along the trajectory, $m > 0$ is the clearance margin, and $\tau > 0$ sets the width of the transition around m. Low-clearance portions of the prediction incur larger penalties. When clearance falls well below m, the penalty grows approximately linearly as the prediction approaches an obstacle. For clearance well above m relative to τ, both the penalty and its slope become small. The penalty encourages predicted trajectories to stay farther from nearby obstacles, with little effect on paths through open space.

To illustrate how the two loss gradients affect trajectory prediction, we consider the two cases in Figure 3. The arrows show negative-gradient updates to the predicted trajectory during training. In Figure 3a, the prediction lies closer to the obstacle than the expert path. The matching loss pulls it toward the expert, while the collision-risk penalty pushes it away from the obstacle. Both local updates favor greater clearance. In Figure 3b, the expert path passes closer to the obstacle than the prediction. Reducing the matching error would move the prediction toward the obstacle, whereas reducing the collision-risk penalty moves it outward. The two objectives therefore compete over how closely to follow the expert near the obstacle. When the expert path passes close to an obstacle,

the collision-risk penalty encourages a prediction with greater clearance. This gives the model an incentive to take a safer path even when doing so increases the trajectory-matching loss.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and data splits. We evaluate GeoRisk-IL on OpenUAV (Wang et al., 2025b) and the obstacle-aware region traversal tasks of HUGE-Bench (Guo et al., 2026). On OpenUAV, Test Seen uses maps and target objects included in the training set. Test Unseen Map (UM) evaluates navigation in maps absent from training, while Test Unseen Object (UO) evaluates navigation to target objects not seen during training. For each split, we report results over all test episodes (Full) and separately for the Easy and Hard subsets. Easy tasks have ground-truth trajectories no longer than 250 m, while Hard tasks have trajectories longer than 250 m. All experimental results reported for OpenUAV are under the L1 assistant setting unless otherwise specified. On HUGE-Bench, we report overall navigation performance and collision rate for the obstacle-aware region traversal tasks.

Evaluation metrics. For OpenUAV, collision rate (CR) is the percentage of trajectories with collision. Navigation error (NE) is the mean Euclidean distance between the UAV's final position and the goal. Success rate (SR) measures the percentage of episodes in which the UAV successfully reaches the goal. Oracle success rate (OSR) measures the percentage of episodes in which the UAV comes within the success distance of the goal at any point along its trajectory. Success weighted by path length (SPL) (Anderson et al., 2018a) measures navigation efficiency by weighting successful episodes according to the executed path length relative to the reference path length. NE is measured in meters, while CR, SR, OSR, and SPL are reported as percentages. Lower CR and NE and higher SR, OSR, and SPL indicate better performance.

For HUGE-Bench, we report SR and CR together with normalized dynamic time warping (nDTW) (Ilharco et al., 2019) and collision-aware success weighted by path length (CSPL). nDTW measures how closely an executed trajectory follows the reference trajectory through an order-preserving alignment of their positions. CSPL combines goal-reaching success, collision-free execution, and path efficiency: an episode receives a nonzero score only if it succeeds without a collision, with longer paths receiving lower scores. SR and CR are expressed as percentages, while nDTW and CSPL are reported on a scale from 0 to 1. Higher nDTW, SR, and CSPL and lower CR indicate better performance.

Baselines. On OpenUAV, we compare GeoRisk-IL with Random Action, CMA (Anderson et al., 2018b), TravelUAV (Wang et al., 2025b), NavFoM (Zhang et al., 2026), LongFly (Jiang et al., 2025), and AirForesight (Liu et al., 2026). We use the results reported in each method's paper; Random Action and CMA follow the OpenUAV benchmark results (Wang et al., 2025b). On HUGE-Bench, the comparison includes π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), and Qwen3-VL Base. We also report Qwen3-VL Base on OpenUAV. This baseline uses expert trajectory supervision without geometric representation alignment or the collision-risk penalty. Section 4.3 examines the two training objectives, and Section 4.4 compares geometric alignment with geometry features supplied as model inputs.

Implementation details. GeoRisk-IL uses Qwen3-VL-2B-Instruct (Bai et al., 2025) as the navigation backbone and the Depth Anything 3 model (Lin et al., 2025) as the geometry teacher. The pretrained VLM weights remain fixed, while the LoRA parameters, trajectory prediction head, and alignment head are optimized during training. The navigation model takes RGB observations, a language instruction, and UAV state as input. Depth observations and UAV poses provide the local obstacle geometry used to compute the training-time collision-risk penalty. The geometry teacher, alignment head and collision-risk penalty computation are used only during training.

4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

Results on OpenUAV. Table 1 summarizes the OpenUAV results. GeoRisk-IL achieves the best NE, SR, OSR, and SPL across the Seen, Unseen Map (UM), and Unseen Object (UO) splits and the Full, Easy, and Hard subsets, with lower CR than Qwen3-VL Base in every group. On Test Seen Full, SR and SPL reach 77.64% and 69.87%, compared with 64.74% and 59.18% for Qwen3-VL Base.

Table 1: Navigation performance on OpenUAV. Seen, UM, and UO denote Test Seen, Test Unseen Map, and Test Unseen Object, respectively. NE is in meters; CR, SR, OSR, and SPL are percentages.

Method	Split	Full					Easy					Hard				
		CR↓	NE↓	SR↑	OSR↑	SPL↑	CR↓	NE↓	SR↑	OSR↑	SPL↑	CR↓	NE↓	SR↑	OSR↑	SPL↑
Random Action	Seen	-	222.20	0.14	0.21	0.07	-	142.07	0.26	0.39	0.13	-	320.12	0.00	0.00	0.00
CMA	Seen	-	135.73	8.37	18.72	7.90	-	84.89	11.48	24.52	10.68	-	197.77	4.57	11.65	4.51
TravelUAV	Seen	-	106.28	16.10	44.26	14.30	-	68.78	18.84	47.61	16.39	-	152.04	12.76	40.16	11.76
NavFoM	Seen	-	93.05	29.17	49.24	25.03	-	58.98	32.91	53.16	27.87	-	143.83	23.58	43.40	20.80
LongFly	Seen	-	60.02	36.39	65.87	31.07	-	38.10	38.52	71.90	31.24	-	85.20	33.94	58.94	30.88
AirForesight	Seen	-	56.99	35.83	69.25	30.22	-	38.27	38.83	71.26	31.65	-	80.08	32.13	66.77	28.45
Qwen3-VL Base	Seen	29.90	51.82	64.74	68.62	59.18	29.50	38.03	64.75	69.09	57.15	30.39	68.83	64.72	68.03	61.68
GeoRisk-IL	Seen	**17.00**	**33.98**	**77.64**	**80.47**	**69.87**	**17.75**	**25.47**	**77.01**	**80.59**	**66.37**	**16.06**	**44.47**	**78.43**	**80.31**	**74.18**
Random Action	UM	-	202.98	0.00	0.00	0.00	-	158.46	0.00	0.00	0.00	-	265.88	0.00	0.00	0.00
CMA	UM	-	141.68	2.30	10.02	2.16	-	102.29	3.57	14.26	3.33	-	197.35	0.50	4.03	0.50
TravelUAV	UM	-	138.80	4.18	20.77	3.84	-	102.94	4.63	22.82	4.24	-	189.46	3.53	17.88	3.28
NavFoM	UM	-	125.10	6.30	18.95	5.68	-	102.41	6.77	20.07	6.04	-	170.58	5.36	15.71	4.97
LongFly	UM	-	108.32	11.27	30.27	9.32	-	78.56	12.96	34.31	10.32	-	148.10	9.02	24.88	7.98
AirForesight	UM	-	90.30	13.67	41.54	12.02	-	67.83	15.86	46.35	13.65	-	122.06	10.58	34.76	9.71
Qwen3-VL Base	UM	63.26	108.96	23.49	27.77	21.70	61.14	76.48	28.16	33.69	25.57	66.25	154.85	16.88	19.40	16.22
GeoRisk-IL	UM	**38.94**	**78.24**	**44.89**	**50.52**	**39.59**	**34.76**	**58.44**	**50.27**	**55.08**	**43.20**	**44.84**	**106.20**	**37.28**	**44.08**	**34.49**
Random Action	UO	-	260.14	0.16	0.16	0.16	-	174.10	0.48	0.48	0.48	-	302.96	0.00	0.00	0.00
CMA	UO	-	155.79	9.06	16.06	8.68	-	102.92	14.83	22.49	13.90	-	182.09	6.19	12.86	6.08
TravelUAV	UO	-	118.11	22.42	46.90	20.51	-	86.12	24.40	49.28	22.03	-	134.03	21.43	45.71	19.75
NavFoM	UO	-	108.04	29.83	47.99	27.20	-	70.51	32.54	50.72	29.54	-	133.01	28.03	46.18	25.64
LongFly	UO	-	66.74	43.87	64.56	38.39	-	54.84	38.01	56.84	31.36	-	57.07	50.25	74.16	45.27
AirForesight	UO	-	66.98	44.36	72.66	39.03	-	50.11	48.80	75.60	41.57	-	75.37	42.14	71.19	37.76
Qwen3-VL Base	UO	25.44	38.27	69.63	78.70	65.00	28.71	36.47	67.46	74.16	60.54	23.81	39.17	70.71	80.95	67.23
GeoRisk-IL	UO	**15.90**	**31.23**	**79.65**	**85.69**	**73.76**	**13.88**	**23.45**	**82.30**	**85.65**	**72.69**	**16.90**	**35.11**	**78.33**	**85.71**	**74.29**

NE decreases from $51.82\,\text{m}$ to $33.98\,\text{m}$, while OSR rises from 68.62% to 80.47%. CR decreases from 29.90% to 17.00%, a reduction of 12.90 percentage points. The gains over Qwen3-VL Base demonstrate the value of geometric representation learning and collision-risk supervision beyond expert trajectory matching.

The improvements extend to both unseen maps and unseen target objects. On UM Full, GeoRisk-IL raises SR from Base's 23.49% to 44.89% and SPL from 21.70% to 39.59%. NE decreases from $108.96\,\text{m}$ to $78.24\,\text{m}$, and CR falls from 63.26% to 38.94%. On UO Full, SR increases from 69.63% to 79.65%, SPL improves from 65.00% to 73.76%, and CR drops from 25.44% to 15.90%.

GeoRisk-IL also improves all five metrics over Base on every Hard subset. UM Hard remains the most challenging setting, but SR rises from 16.88% to 37.28% and CR decreases from 66.25% to 44.84%. On Seen Hard and UO Hard, SR reaches 78.43% and 78.33%, respectively, with SPL scores of 74.18% and 74.29%.

Results on HUGE-Bench. Table 2 reports results on obstacle-aware region traversal in HUGE-Bench. GeoRisk-IL achieves the highest SR and CSPL and the lowest CR. Compared with Qwen3-VL Base, SR increases from 40.96% to 44.58%, and CR decreases from 24.10% to 16.87%, a 30.0% relative reduction.

CSPL improves from 0.3974 to 0.4213. GeoRisk-IL also outperforms both π_0 and $\pi_{0.5}$ on all four metrics, with an SR gain of 7.23 percentage points.

Table 2: Overall performance on HUGE-Bench obstacle-aware region traversal tasks. SR/CR are percentages; best values are bold.

Method	nDTW↑	SR↑	CR↓	CSPL↑
π_0	0.1645	37.35	20.48	0.3608
$\pi_{0.5}$	0.1475	37.35	40.96	0.3253
Qwen3-VL Base	**0.1958**	40.96	24.10	0.3974
GeoRisk-IL	0.1940	**44.58**	**16.87**	**0.4213**

nDTW changes only slightly from Base to GeoRisk-IL (0.1958 to 0.1940). GeoRisk-IL nevertheless completes more episodes successfully and has fewer collisions.

4.3 Ablation Study

We study the contributions of the two auxiliary training objectives, geometric representation alignment and the collision-risk penalty, on the full OpenUAV Test Seen set. All variants use the same model architecture and expert trajectory-matching supervision; they differ only in whether $\mathcal{L}_{\text{geo}}$ and $\mathcal{L}_{\text{risk}}$ are enabled.

Table 3: Ablation study on OpenUAV Test Seen. NE: meters; CR/SR/OSR/SPL: percentages.

	Training supervision		Navigation performance				
Variant	Geometry	Collision	CR ↓	NE ↓	SR ↑	OSR ↑	SPL ↑
Qwen3-VL Base	×	×	29.90	51.82	64.74	68.62	59.18
Geometry only	✓	×	29.90	51.23	64.46	67.35	58.97
Penalty only	×	✓	21.44	40.63	71.30	75.39	64.13
GeoRisk-IL	✓	✓	**17.00**	**33.98**	**77.64**	**80.47**	**69.87**

Table 3 compares four variants. Qwen3-VL Base uses trajectory supervision alone. *Geometry only* adds geometric representation alignment, *Penalty only* adds the collision-risk penalty, and GeoRisk-IL combines both objectives. Geometry alignment alone has little effect on navigation performance: CR remains at 29.90%, while SR changes from 64.74% to 64.46%. Adding the collision-risk penalty alone reduces CR from 29.90% to 21.44% and increases SR from 64.74% to 71.30%, with NE decreasing from 51.82 to 40.63 and SPL increasing from 59.18% to 64.13%.

The collision-risk penalty produces larger gains when geometric alignment is also used. Adding the penalty to the Geometry-only variant improves SR by 13.18 percentage points, approximately twice the 6.56-point gain obtained when adding it to Base. The corresponding CR reduction is also larger: 12.90 percentage points with geometric alignment, compared with 8.46 points without it. With both objectives enabled, GeoRisk-IL achieves an SR of 77.64% and a CR of 17.00%, while also obtaining the best NE, OSR, and SPL among the four variants. Compared with Base, this corresponds to a 43.16% relative reduction in collision episodes.

These results indicate that geometric alignment is most useful when combined with explicit collision-risk supervision. The alignment objective encourages the visual representations to encode scene geometry, while the collision-risk penalty uses obstacle clearance to supervise the model's predicted trajectories. Their combination helps the RGB-only navigation model make more effective use of the risk signal during training.

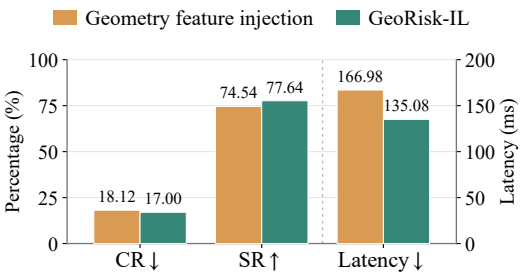

Figure 4: Geometry supervision vs. geometry feature injection on OpenUAV Test Seen.

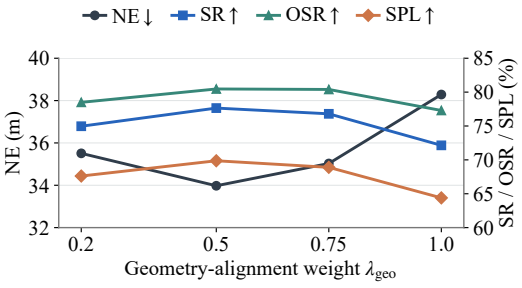

Figure 5: Effect of geometry-alignment weight on OpenUAV Test Seen.

4.4 Variant Study

Training-Time Geometry Supervision vs. Geometry Feature Injection. We compare two ways of incorporating pretrained geometric information into the navigation model. The *Geometry feature injection* variant runs DA3 at every navigation step and feeds its projected features to the LLM as additional input tokens. In contrast, GeoRisk-IL uses DA3 features only during training through

geometric representation alignment; DA3 and the alignment head are removed at inference. Both variants retain the collision-risk penalty.

As shown in Figure 4, the Geometry feature injection variant achieves a CR of 18.12% and an SR of 74.54%, with an inference latency of 166.98 ms. GeoRisk-IL reduces CR to 17.00% and increases SR to 77.64%, while reducing latency to 135.08 ms, a 19.10% reduction. These results show that training-time geometric alignment can transfer useful spatial information to the navigation model without requiring DA3 during inference. Removing online DA3 execution and additional feature-token processing also reduces inference cost.

4.5 Parameter Analysis

Geometry-alignment weight. The coefficient λ_{geo} controls the contribution of the geometric representation alignment loss to the overall training objective. We study its effect by varying $\lambda_{\text{geo}} \in \{0.2, 0.5, 0.75, 1.0\}$ on the full OpenUAV Test Seen split.

Figure 5 shows that a moderate alignment weight gives the best navigation performance. Increasing λ_{geo} from 0.2 to 0.5 raises SR from 74.96% to 77.64%, and $\lambda_{\text{geo}} = 0.5$ also gives the lowest NE and the highest OSR and SPL. Increasing the weight to 0.75 provides no further improvement, while setting it to 1.0 degrades all four metrics. This trend indicates that geometric alignment is most effective as an auxiliary supervision signal with a moderate weight. A small weight provides limited geometric supervision, whereas an overly large weight can place too much emphasis on matching the geometry teacher features and interfere with features needed for trajectory prediction.

4.6 Qualitative Analysis

Figure 6 compares six consecutive front-view observations of Qwen3-VL Base and GeoRisk-IL, starting from the same position. Qwen3-VL Base flies into a tree and becomes stuck among the branches. GeoRisk-IL climbs above the trees and continues toward the target without collision.



Figure 6: Qualitative comparison between Qwen3-VL Base and GeoRisk-IL on OpenUAV. Both runs start at the same position. Qwen3-VL Base (top) collides with the trees, while GeoRisk-IL (bottom) gains altitude and flies above them.

5 Conclusion

We presented GeoRisk-IL for safer UAV vision-language navigation. Expert demonstrations vary in obstacle clearance, but trajectory matching alone does not explicitly account for these differences. GeoRisk-IL penalizes predicted trajectories that approach obstacles and introduces geometric representation alignment to improve spatial understanding. Experiments on OpenUAV and HUGE-Bench show higher navigation success rate and fewer collisions. Geometric alignment provides little benefit on its own, yet substantially increases the gains from the collision-risk penalty. These findings support using geometric learning alongside explicit risk supervision, while retaining RGB-only inference without additional geometric modules.

AI USE STATEMENT

Generative AI tools were used to improve the readability and language of the manuscript.

ETHICS STATEMENT

The research conducted in this paper conforms to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The code is available through the link at the end of the abstract.

REFERENCES

Maximilian Adang, JunEn Low, Ola Shorinwa, and Mac Schwager. Singer: An onboard generalist vision-language navigation policy for drones. *arXiv preprint arXiv:2509.18610*, 2025.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3674–3683. IEEE, 2018b.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Valts Blukis, Dipendra Misra, Ross A Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. In *Conference on Robot Learning*, pp. 505–518. PMLR, 2018.

Hengxing Cai, Jinhan Dong, Jingjun Tan, Jingcheng Deng, Sihang Li, Zhifeng Gao, Haidong Wang, Zicheng Su, Agachai Sumalee, and Renxin Zhong. Flightgpt: Towards generalizable and interpretable uav vision-and-language navigation with vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6670–6687, 2025.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16516–16526. IEEE, 2022.

Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Wang. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3043–3061, 2023.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Jingyu Guo, Ziye Chen, Ziwen Li, Zhengqing Gao, Jiaxin Huang, Hanlue Zhang, Fengming Huang, Yu Yao, Tongliang Liu, and Mingming Gong. Huge-bench: a benchmark for high-level uav vision-language-action tasks. In *European Conference on Computer Vision*, pp. 373–388. Springer, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615. IEEE, 2023.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

Wen Jiang, Li Wang, Kangyao Huang, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hongwei Duan, Bin Xu, and Xiangyang Ji. Longfly: Long-horizon uav vision-and-language navigation with spatiotemporal context integration. *arXiv preprint arXiv:2512.22010*, 2025.

Wen Jiang, Kangyao Huang, Li Wang, Wang Xu, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hanfang Liang, Hongwei Duan, Bin Xu, et al. Spatialfly: Geometry-guided representation alignment for uav vision-and-language navigation in urban environments. *arXiv preprint arXiv:2603.21046*, 2026a.

Wen Jiang, Hanfang Liang, Li Wang, Kangyao Huang, Wang Xu, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hongwei Duan, Bin Xu, et al. Dynfly: Dynamic-aware continuous trajectory generation for uav vision-language navigation in urban environments. *arXiv preprint arXiv:2606.31654*, 2026b.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: A large-scale dataset for real-world aerial navigation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5912–5922. IEEE, 2025.

Guoxin Lian, Shuo Wang, Yucheng Wang, Yongcai Wang, Maiyue Chen, Kaihui Wang, Bo Zhang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Mapdream: Task-driven map learning for vision-language navigation. *arXiv preprint arXiv:2602.00222*, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Rui Liu, Wenguan Wang, and Yi Yang. Volumetric environment representation for vision-language navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16317–16328. IEEE, 2024.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15338–15348. IEEE, 2023.

Yutong Liu, Xiaojie Li, Mingzhu Xu, and Jianlong Wu. Airforesight: Current-to-future spatial map imagination with cross-space planning consistency for uav-vln. *arXiv preprint arXiv:2608.12835*, 2026.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018.

Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. *arXiv preprint arXiv:2103.03454*, 2021.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Hongsheng Li, Yue Liao, Si Liu, et al. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology. In *International Conference on Learning Representations*, volume 2025, pp. 7292–7310, 2025b.

Xiangyu Wang, Donglin Yang, Yue Liao, Wenhao Zheng, Bin Dai, Hongsheng Li, Si Liu, et al. Uav-flow colosseo: A real-world benchmark for flying-on-a-word uav imitation learning. *Advances in Neural Information Processing Systems*, 38, 2026.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13753–13762, 2024.

Jiahao Yang, Zihan Wang, Xiangyang Li, Xing Zhu, Yujun Shen, Yinghao Xu, and Shuqiang Jiang. Ga-vln: Geometry-aware bev representation for efficient vision-language navigation. *arXiv preprint arXiv:2605.22036*, 2026.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056, 2025.

A Appendix

A.1 OpenUAV Data and Optimization

The training index contains 427,933 observations from 7,922 expert trajectories across 19 environments. Each observation uses the current front and down RGB images at 256×256 resolution.

Table 4: OpenUAV training configuration for GeoRisk-IL.

Setting	Value
Navigation backbone	Qwen3-VL-2B-Instruct
Geometry teacher	DA3-LARGE-1.1, frozen
Epochs	2
GPUs / batch per GPU	4 / 32
Optimizer	AdamW; $(\beta_1, \beta_2) = (0.9, 0.999)$
Learning rate / weight decay	5×10^{-5} / 0
Optimizer ϵ / gradient clipping norm	10^{-8} / 1.0
Schedule / learning-rate warmup	Cosine / 3% of main updates
λ_{geo} / λ_{risk}	0.5 / 10
Stopping-loss weight	0.1

LoRA adapts the attention and feed-forward projections in the language backbone and visual encoder. Original language weights remain frozen. The task heads are also trained.

The model predicts ten trajectory points, and the UAV follows the first five before the next observation.

A.2 Navigation and Alignment Losses

For a mini-batch $\mathcal{B}$, let v_t^k indicate whether future trajectory point k is valid at observation t. The trajectory matching loss is

$$\mathcal{L}_{\text{traj}} = \frac{\sum_{t \in \mathcal{B}} \sum_{k=1}^{K} v_t^k \sum_{d=1}^{3} \ell_{\text{SL1}}\big(\hat{w}_{t,d}^k - w_{t,d}^{k,*}\big)}{3 \sum_{t \in \mathcal{B}} \sum_{k=1}^{K} v_t^k}. \tag{6}$$

Here, ℓ_{SL1} denotes the Smooth L1 loss. All valid steps and coordinate dimensions receive equal weight. Equation 6 averages valid coordinate losses within each worker's mini-batch; distributed training averages gradients across workers. A batch with no valid future trajectory points contributes zero trajectory loss.

For geometric alignment, DA3 processes the current front/down pair jointly. Its input images are resized to 252×252. We extract the last exported feature layer (index 23) and use adaptive average pooling to obtain an 8×8 grid of 1,024-dimensional features per view. The student features are the 2,048-dimensional hidden states at the corresponding image-token positions after the final LLM layer. The alignment head consists of LayerNorm, a $2048 \rightarrow 2048$ linear layer, GELU, and a $2048 \rightarrow 1024$ linear layer.

A.3 Computing the Collision-Risk Penalty

Starting at the current position $\hat{w}_t^0$, we linearly interpolate the segments joining the first $K_e = 5$ predicted trajectory points. Each segment contributes $S = 4$ positions:

$$\mathbf{q}_t^{k,j} = \hat{w}_t^{k-1} + \frac{j}{S}(\hat{w}_t^k - \hat{w}_t^{k-1}), \qquad k = 1, \ldots, K_e, \quad j = 1, \ldots, S. \tag{7}$$

The positions are equally spaced within each segment and include its endpoint, giving 20 positions for a complete execution prefix. The current position itself is not included in the loss.

Let $\mathcal{Q}_t$ contain the valid interpolated positions. We evaluate the clearance $c_t(\mathbf{q})$ from Eq. 4 at each position and average the penalty first over $\mathcal{Q}_t$, then over the mini-batch:

$$\mathcal{L}_{\text{risk}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \frac{1}{|\mathcal{Q}_t|} \sum_{\mathbf{q} \in \mathcal{Q}_t} \frac{\tau}{m} \log \left(1 + \exp \left(\frac{m - c_t(\mathbf{q})}{\tau} \right) \right). \tag{8}$$

We use $m = 2\,\text{m}$ and $\tau = 0.25\,\text{m}$.

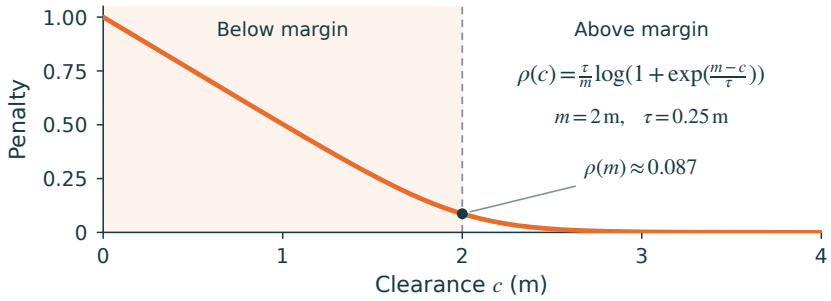

Figure 7: Collision-risk penalty in Eq. 5, illustrated with $m = 2\,\text{m}$ and $\tau = 0.25\,\text{m}$. The penalty decreases smoothly with clearance c.

A.4 Stop Supervision

A separate head learns when to end navigation. It reads the final hidden state h_t^{stop} at the <stop> token and maps it to a scalar with a two-layer MLP. The stopping probability is

$$p_t^{\text{stop}} = \sigma\big(f_{\text{stop}}(h_t^{\text{stop}})\big). \tag{9}$$

The training target depends on d_t, the Euclidean distance in meters from the current UAV position to the endpoint of the expert trajectory:

$$y_t^{\text{stop}} = \sigma\left(\frac{20 - d_t}{5} \right). \tag{10}$$

This target rises smoothly as the UAV approaches the endpoint, with a midpoint at 20 m. We fit the predicted probability using binary cross-entropy over the observations with valid stopping labels, $\mathcal{B}_{\text{stop}}$:

$$\mathcal{L}_{\text{stop}} = -\frac{1}{|\mathcal{B}_{\text{stop}}|} \sum_{t \in \mathcal{B}_{\text{stop}}} \left[y_t^{\text{stop}} \log p_t^{\text{stop}} + (1 - y_t^{\text{stop}}) \log(1 - p_t^{\text{stop}}) \right]. \tag{11}$$

At inference, the stop prediction is activated at $p_t^{\text{stop}} \geq 0.85$. For OpenUAV, we set $\lambda_{\text{stop}} = 0.1$, $\lambda_{\text{risk}} = 10$, and $\lambda_{\text{geo}} = 0.5$. Substituting these weights into Equation 1 gives

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + 0.1\mathcal{L}_{\text{stop}} + 10\mathcal{L}_{\text{risk}} + 0.5\mathcal{L}_{\text{geo}}. \tag{12}$$

A.5 HUGE-Bench Adaptation

HUGE-Bench uses the current front image and predicts 20 local motion increments, each containing translation and yaw. The first ten increments are executed before replanning. Action targets are standardized using training-set statistics. Both methods use the same action-regression objective for $\mathcal{L}_{\text{nav}}$:

$$\mathcal{L}_{\text{nav}}^{\text{HUGE}} = \mathcal{L}_{\text{step}} + 0.5\mathcal{L}_{\text{cum}} + \mathcal{L}_{\text{end10}} + 0.25\mathcal{L}_{\text{end20}} + 0.05\mathcal{L}_{\text{delta}} + \mathcal{L}_{\text{static}}. \tag{13}$$

Smooth L1 terms match normalized increments, cumulative poses, the poses after ten and twenty increments, and differences between adjacent increments. Translation errors are averaged over coordinates; yaw residuals are wrapped to $[-\pi, \pi]$. Each term is averaged over its valid targets.

For GeoRisk-IL, the geometry-alignment and collision-risk loss weights remain 0.5 and 10, respectively, with a 2 m clearance margin. Both methods are evaluated after five training epochs. Training uses four GPUs, batch size 16 per GPU, and no gradient accumulation. LoRA uses rank 16, $\alpha = 32$, and dropout 0.05. The LoRA and head learning rates are 5×10^{-5} and 10^{-4}, respectively. AdamW uses weight decay 0.01, with a 3% learning-rate warmup.